

LADAS & PARRY  
5670 Wilshire Boulevard  
Suite 2100  
Los Angeles, CA 90036  
Telephone: (323) 934-2300  
Telefax: (323) 934-0202

**THIS PAGE BLANK (USP<sup>™</sup>)**

USIN 10 632,123



INVESTOR IN PEOPLE

The Patent Office  
Concept House  
Cardiff Road  
Newport  
South Wales  
NP10 8QQ

**BEST AVAILABLE COPY**

11,632,123

**CERTIFIED COPY OF  
PRIORITY DOCUMENT**

, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

accordance with the Patents (Companies Re-registration) Rules 1982, if a company named on this certificate and any accompanying documents has re-registered under the Companies Act 2006 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company on this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., p.l.c. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

*Evans*

Dated

18 October 2004

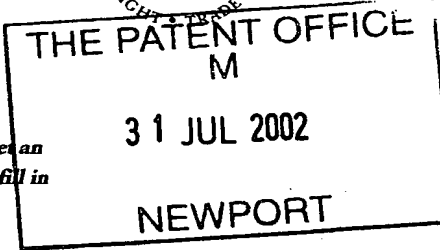
**THIS PAGE BLANK (USPTO)**



01AUG02 E737672-1 D01463  
P01/7700-0.00-0217791.3

# Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)



The Patent Office

Cardiff Road  
Newport  
South Wales  
NP10 8QQ

1. Your reference	30016998-1 GB		
2. Patent application number (The Patent Office will fill in this part)	0217791.3		31 JUL 2002
3. Full name, address and postcode of the or of each applicant (underline all surnames)	Hewlett-Packard Company 3000 Hanover Street Palo Alto CA 94304, USA  Delaware, USA		
Patents ADP number (if you know it)	496588001		
If the applicant is a corporate body, give the country/state of its incorporation	Delaware, USA		
4. Title of the invention	Identifying Uncorrectable Codewords in a Reed-Solomon Decoder for Errors and Erasures		
5. Name of your agent (if you have one)	Richard A. Lawrence Hewlett-Packard Ltd, IP Section Filton Road, Stoke Gifford Bristol BS34 8QZ		
"Address for service" in the United Kingdom to which all correspondence should be sent (including the postcode)	7563083001		
Patents ADP number (if you know it)			
6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (if you know it) the or each application number	Country	Priority application number (if you know it)	Date of filing (day / month / year)
7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application	Number of earlier application	Date of filing (day / month / year)	
8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (Answer 'Yes' if:	Yes		
a) any applicant named in part 3 is not an inventor, or			
b) there is an inventor who is not named as an applicant, or			
c) any named applicant is a corporate body.			
See note (d))			

## Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document.

Continuation sheets of this form

Description

20 ✓

Claim(s)

3 ✓

Abstract

1 ✓ *h*

Drawing(s)

2 + 2 ✓

10. If you are also filing any of the following, state how many against each item.

Priority documents

-

Translations of priority documents

-

Statement of inventorship and right to grant of a patent (*Patents Form 7/77*)

1 ✓

Request for preliminary examination and search (*Patents Form 9/77*)

1 ✓

Request for substantive examination (*Patents Form 10/77*)

-

Any other documents  
(*please specify*)

Fee Sheet

11.

I/We request the grant of a patent on the basis of this application.

Signature

Richard A. Lawrence

Date

30/7/02

12. Name and daytime telephone number of person to contact in the United Kingdom

Meg Joyce Tel: 0117-312-9068

### Warning

*After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.*

### Notes

- If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*
- Write your answers in capital letters using black ink or you may type them.*
- If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*
- If you have answered 'Yes' Patents Form 7/77 will need to be filed.*
- Once you have filled in the form you must remember to sign and date it.*
- For details of the fee and ways to pay please contact the Patent Office.*

EX-100-1

IDENTIFYING UNCORRECTABLE CODEWORDS IN A  
REED-SOLOMON DECODER FOR ERRORS AND ERASURES

5       The present invention relates in general to a Reed-Solomon decoder. More particularly, the invention relates to a method and apparatus for identifying uncorrectable codewords in the presence of errors and erasures.

10

Reed-Solomon RS error correcting codes are applicable to a wide variety of environments. The most notable examples are data storage, such as on optical disc media or solid state media, and long-distance data  
15 transmissions, such as from deep space to earth.

RS encoded data comprises codewords each having a predetermined number of symbols. When RS encoded data is stored or transmitted, the value of one or more symbols in  
20 a codeword may become corrupt, i.e. become changed to an incorrect value or become unknown. This is termed a symbol error. In some environments, the location of a corrupt symbol can be predicted and declared in advance of decoding (e.g. because data could not be read from a  
25 storage media, or part of a data transmission was not received). This is termed a symbol erasure. The term errata refers generally to either errors or erasures.

Most RS decoders handle only errors. However, some  
30 known RS decoders are able to perform decoding handling both errors and erasures. Here, the decoder receives erasure information, which indicates the location of zero or more suspected corrupt symbols within a codeword.

Conveniently, a one bit erasure indicator is applied to each symbol, to indicate whether that symbol has been declared as an erasure. An RS  $(B, B-2T)$  code forms codewords of length  $B$  and is mathematically guaranteed to  
5 correct up to  $T$  symbol errors, and up to  $2T$  symbols erasures, or any pattern of  $E$  symbol errors and  $J$  symbol erasures, provided that  $2E + J \leq 2T$ . To correct an error, the decoder must find both its location and its magnitude, whereas to correct an erasure, only the magnitude of the  
10 correction must be found. Hence, erasure information allows more corrupt symbols to be corrected, compared with the situation where the erasure information is not available.

15 When a received codeword is passed to the decoder for decoding, the number of declared erasures is known. However, the number of errors is not known and must be determined by the decoder. On occasion, a codeword is encountered which is so severely corrupted that the  
20 decoder is unable to provide a reliable, mathematically accurate, correction, i.e. the received codeword is uncorrectable. It is important in many practical environments to determine that an uncorrectable codeword has been received.

25

An example prior art apparatus for detecting uncorrectable error patterns is disclosed in US 5,099,482 (Idaho Research Foundation). However, this prior art apparatus is not applicable to a Reed-Solomon decoder that  
30 is able to handle both errors and erasures.

An aim of the present invention is to provide a method and apparatus for identifying uncorrectable Reed-Solomon



codewords in the presence of errors and erasures. A preferred aim is to distinguish between correctable and uncorrectable codewords accurately and cost-efficiently.

5       According to a first aspect of the present invention there is provided a method of identifying uncorrectable codewords in a Reed-Solomon decoder handling errors and erasures, comprising the steps of: indicating an uncorrectable codeword when any one or more of the  
10       following conditions (a) to (f) is satisfied:

- (a) no solution to key equation  $\sigma(x)T(x) \equiv \omega(x) \bmod x^{2T}$ ;
- (b)  $\deg \sigma(x) \neq \text{nerrors}$ ;
- (c) error and erasure locations coincide;
- 15       (d)  $\deg \omega(x) \geq \text{nerrors} + \text{nerasures}$ ;
- (e)  $\text{nerasures} + 2 * \text{nerrors} > 2T$ ; and
- (f) an error location has a zero correction magnitude;

where  $\text{nerrors}$  and  $\text{nerasures}$  represent, respectively, a  
20       number of errors with reference to an error locator polynomial  $\sigma(x)$  and a number of erasures with reference to an erasure locator polynomial  $\Lambda(x)$ ,  $2T$  is the strength of a Reed-Solomon code,  $\omega(x)$  is an errata evaluator polynomial, and  $T(x)$  is a modified syndrome polynomial.

25

Preferably the method comprises evaluating the condition (a) as a preliminary step, and then evaluating the conditions (b) to (f).

30       Preferably, the method comprises identifying a codeword as correctable if none of at least the conditions (a) to (f) are satisfied.

Optionally, the method comprises indicating an uncorrectable codeword in response to condition (g)  $\deg\Lambda(x) \neq \text{nerasures}$ .

5 Preferably, the method comprises receiving the error locator polynomial  $\sigma(x)$ , the erasure locator polynomial  $\Lambda(x)$  and the errata evaluator polynomial  $\omega(x)$ ; forming a set of error locations, and a set of erasure locations, and forming variables  $\text{nerrors}$  and  $\text{nerasures}$  representing  
 10 the size of each set, respectively; and finding  $\deg\sigma(x)$ ,  $\deg\Lambda(x)$ , and  $\deg \omega(x)$ , as a degree of the error locator polynomial  $\sigma(x)$ , the erasure locator polynomial  $\Lambda(x)$  and the errata evaluator polynomial  $\omega(x)$ , respectively.

15 According to a second aspect of the present invention there is provided a detector circuit arranged to identify an uncorrectable codeword, for use in Reed-Solomon decoder handling errors and erasures, the circuit comprising: a logic unit arranged to identify each condition:

- 20
- (a) no solution to key equation  $\sigma(x)T(x) \equiv \omega(x) \bmod x^{2T}$ ;
  - (b)  $\deg\sigma(x) \neq \text{nerrors}$ ;
  - (c) error and erasure locations coincide;
  - (d)  $\deg\omega(x) \geq \text{nerrors} + \text{nerasures}$ ;
  - 25 (e)  $\text{nerasures} + 2*\text{nerrors} > 2T$ ; and
  - (f) an error location has a zero correction magnitude;

where  $\text{nerrors}$  and  $\text{nerasures}$  represents, respectively, a number of errors and erasures with reference to an error  
 30 locator polynomial  $\sigma(x)$  and an erasure locator polynomial  $\Lambda(x)$ ,  $2T$  is the strength of a Reed-Solomon code,  $\omega(x)$  is an errata evaluator polynomial, and  $T(x)$  is a modified syndrome polynomial; and an indicator unit arranged to

indicate an uncorrectable codeword, responsive to the logic unit.

Preferably, the circuit comprises a counter arranged  
 5 to count nerrors and nerasures as the size of a set of error locations derived from the error locator polynomial  $\sigma(x)$ , and a set of erasure locations derived from the erasure locator polynomial  $\Lambda(x)$ , respectively.

10 Optionally, the logic unit is arranged to identify an uncorrectable codeword in response to condition (g)  $\deg \Lambda(x) \neq \text{nerasures}$ .

For a better understanding of the invention, and to  
 15 show how embodiments of the same may be carried into effect, reference will now be made, by way of example, to the accompanying diagrammatic drawings in which:

Figure 1 is a schematic block diagram of a preferred  
 20 Reed-Solomon decoder; and

Figure 2 is a schematic circuit diagram of a circuit for identifying uncorrectable codewords.

25 Reed-Solomon (RS) codes are a special subclass of generalised BCH codes. A Reed-Solomon code of the form RS  $(B, B-2T, T)$  defined in a Galois field  $GF(2^w)$  forms codewords of length  $B \leq 2^w - 1$ , where  $w$  is a positive integer. In each codeword of length  $B$  symbols,  $B-2T$  of  
 30 these symbols are information symbols, and the remaining  $2T$  symbols are check symbols.

The generator polynomial  $g(x)$  of an RS code can be expressed in the form:

$$\begin{aligned} g(x) &= (x + \alpha^L)(x + \alpha^{L+1})(x + \alpha^{L+2}) \dots (x + \alpha^{L+2T-1}) \\ &= g_0 + g_1x + g_2x^2 + \dots + g_{2T-1}x^{2T-1} + x^{2T} \end{aligned}$$

5

where  $\alpha$  is a primitive element in  $GF(2^w)$  and  $L$  is an integer constant. Different generating polynomials are formed with different values for  $L$ . By carefully choosing the constant  $L$ , the circuit complexity of the encoder and  
10 decoder can be reduced. In most practical cases it is convenient to let  $L = 1$ .

When original data is encoded using an RS encoder employing the generator polynomial  $g(x)$ , an original  
15 codeword  $c(x)$  is produced which can be represented by the polynomial:

$$c(x) = c_{B-1}x^{B-1} + c_{B-2}x^{B-2} + \dots + c_2x^2 + c_1x + c_0$$

20 Let an error pattern  $e(x)$ , caused for example during data storage or data transmission, be represented by the polynomial:

$$e(x) = e_{B-1}x^{B-1} + e_{B-2}x^{B-2} + \dots + e_2x^2 + e_1x + e_0$$

25

Let a received codeword  $d(x)$  be represented by the polynomial:

$$\begin{aligned} d(x) &= d_{B-1}x^{B-1} + d_{B-2}x^{B-2} + \dots + d_2x^2 + d_1x + d_0 \\ &= c(x) + e(x) \end{aligned}$$

30

The received codeword  $d(x)$  may be perfect, or may be corrupted. Hence, it is desired to check and, if necessary, correct the received codeword, using an RS decoder.

5

A Reed-Solomon decoder can be implemented in software, or in hardware, or as a combination of software and hardware. The RS decoder described herein is particularly suitable for implementation in hardware using VLSI  
10 techniques to form an ASIC (Application Specific Integrated Circuit).

A preferred Reed-Solomon decoder will first be described, with reference to Figure 1. Then, a preferred  
15 circuit for detecting uncorrectable codewords will be described, with reference to Figure 2.

The preferred decoder of Figure 1 comprises a polynomial generation unit 30, a polynomial evaluation  
20 unit 50, and a transform unit 42.

A codeword received by the decoder 10 is passed to the polynomial generation unit 30, which generates polynomials representing locations and magnitudes of errors and  
25 erasures. In this embodiment the polynomial generation unit 30 generates an error locator polynomial  $\sigma(x)$ , an erasure locator polynomial  $\Lambda(x)$  and an errata evaluator polynomial  $\omega(x)$ . These three polynomials are passed to the polynomial evaluation unit 50, which performs a Chien  
30 search and solves Forney's equations to determine correction locations and correction magnitudes to be applied to the received codeword.

Referring to Figure 1, the preferred polynomial generation unit 30 and polynomial evaluation unit 50 will now be described in more detail.

5       The polynomial generation unit 30 comprises a syndrome calculation block 12, an erasurelist block 14, a polynomial expander 22, a key equation solving unit 32, and a delay 34.

10       The syndrome calculation block 12 is arranged to receive a codeword containing B symbols. The symbols of the codeword form the coefficients of a polynomial, where the first symbol received is  $d_{B-1}$  and the last symbol received is  $d_0$ . The received codeword can be represented  
15 as:

$$d(x) = d_0 + d_1x + d_2x^2 + d_3x^3 + \dots + d_{B-1}x^{B-1}$$

20       The syndromes are obtained by evaluating this polynomial at the roots of the generator polynomial. The generator polynomial has  $2T$  distinct roots  $(\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2T})$ , and therefore there are  $2T$  syndromes to be calculated. This assumes that  $L$ , where  $\alpha^L$  is the starting root of the generator polynomial  $g(x)$ , is given  
25 by  $L=1$ , but other values of  $L$  are also applicable. Suitably, the calculation of each syndrome is performed recursively, using Horner's rule.

30       The erasurelist block 14 receives erasure information identifying zero or more symbols in the codeword which have been declared as erasures. Any suitable mechanism can

be used to declare erasures, according to the environment in which the RS decoder 10 is employed.

The erasurelist block 14 is formed as a FIFO structure to maintain a list of up to  $2T$  erasure locations. If the first symbol of the codeword is an erasure, then a value of  $\alpha^{-(B-1)}$  is queued in the FIFO. If the next symbol is an erasure, then  $\alpha^{-(B-2)}$  queued in the FIFO, and so on.

In the architecture of the preferred RS decoder, only a single polynomial expander 22 is provided which calculates both (a) the erasure locator polynomial  $\Lambda(x)$ , and (b) the modified syndrome polynomial  $T(x)$ . In both cases, the same basic operation is used:

15

$$polyout(x) = polyin(x) \cdot (x + \alpha^{-v_0})(x + \alpha^{-v_1})(x + \alpha^{-v_2}) \cdots (x + \alpha^{-v_{j-1}})$$

To calculate  $T(x)$ , the initial value loaded into  $polyin(x)$  is  $S(x)$ . To calculate  $\Lambda(x)$ , the initial value loaded into  $polyin(x)$  is 1.

Conveniently, the same hardware is used to generate both  $T(x)$  and  $\Lambda(x)$ , by time multiplexing these two functions of the polynomial expander 22. That is, in a first mode the polynomial expander 22 generates  $T(x)$  and passes the result to the key equation block 32. Then, in a second mode, the polynomial expander 22 generates  $\Lambda(x)$ .

The key equation block 32 receives the modified syndrome polynomial  $T(X)$  from the polynomial expander 22, and from  $T(x)$  calculates values of the error locator

polynomial  $\sigma(x)$  and the errata evaluator polynomial  $\omega(x)$  that solve the key equation:

$$\sigma(x) \cdot T(x) \equiv \omega(x) \bmod x^{2T}$$

5

The key equation block 32 preferably applies Euclid's algorithm to solve the key equation. In alternate embodiments of the invention, the key equation block 32 is replaced by a functional equivalent. For example, alternate approaches to solving the key equation and finding  $\sigma(x)$  and  $\omega(x)$  include Berlekamp, Berlekamp-Massey, and continued fractions.

A detailed discussion as background to the preferred implementation of Euclid's algorithm is provided in the paper "A Hypersystolic Reed-Solomon Decoder" by E. Berlekamp, G. Seroussi and P. Tong published at pages 205-241 of "Reed-Solomon Codes and their Applications", Edited by S.B. Wicker and V.K. Bhargava, IEEE Press, New York, 1994, ISBN 0-7803-1025-X.

The latency through the key equation block varies depending on the number of erasures  $J$ . The delay block 34 is provided to ensure that the decoder as a whole has constant latency. The delay block 34 indicates that a valid output of the key equation block 32 is available a constant number of cycles after the Euclidean computation starts, rather than on its completion.

The polynomial evaluation unit 50 comprises first, second and third polynomial evaluators, 52, 54 and 56, a



Forney block 62, a correction block 72, and a monitor block 82.

Three polynomial evaluators 52, 54, and 56 are provided to evaluate  $\sigma(x)$ ,  $\Lambda(x)$  and  $\omega(x)$ , respectively. The first polynomial evaluator 52 is used to determine the roots of the error locator polynomial  $\sigma(x)$ , which indicate the locations of symbol errors. An exhaustive search is used to determine these roots, suitably a Chien search or equivalent. Similarly, the second polynomial evaluator 54 performs an equivalent function for the erasure locator polynomial  $\Lambda(x)$ . The third polynomial evaluator 56 evaluates the errata evaluator polynomial  $\omega(x)$ .

Optionally, the transform block 42 is provided to implement a polynomial transformation which allows a Chien search to be performed immediately, and avoids a delay corresponding to "empty" information symbols in a shortened codeword of length  $B \cdot 2^w - 1$ . Here, the  $i^{\text{th}}$  coefficient of an input polynomial is transformed by a factor of  $\alpha^{(2^w - B)i}$ . There are three polynomials that require this transformation, namely the error locator polynomial  $\sigma(x)$  and the errata evaluator polynomial  $\omega(x)$  from the key equation block 32, and the erasure locator polynomial  $\Lambda(x)$  from the polynomial expander block 22. Once a control signal "start" is asserted, these polynomials are processed over successive clock cycles. Three "done" signals indicate to subsequent blocks when each transformed polynomial (and its true degree) is available.

The Forney block 62 is implemented as a pipelined datapath, driven by the three polynomial evaluation blocks

52, 54, 56 for  $\sigma(x)$ ,  $\Lambda(x)$  and  $\omega(x)$ . There is one cycle skew between each of these blocks, due to the transform block 42, and thus the results feed into the datapath at different stages.

5

As described previously, a Chien search (or equivalent) is performed by the polynomial evaluators 52, 54, 56 evaluating  $\sigma(x)$ ,  $\omega(x)$  and  $\Lambda(x)$  for  $x \in \{\alpha^{-(B-1)} \dots \alpha^{-(0)}\}$ . From these values, Forney equations are used in the Forney  
10 block 62 to calculate the error magnitudes.

In the preferred embodiment, the Forney equations are applied in the form that if  $\sigma(x)=0$  for some  $x=\alpha^{-i}$  then an error has occurred in symbol  $i$ , and the error magnitude is  
15 given by:

$$E_i = \frac{\omega(x)}{\sigma'(x) \cdot \Lambda(x)} \text{ for } x = \alpha^{-i};$$

and if  $\Lambda(x)=0$  for some  $x=\alpha^{-i}$  then an erasure has occurred  
20 in symbol  $i$ , and the erasure magnitude is given by:

$$J_i = \frac{\omega(x)}{\sigma(x) \cdot \Lambda'(x)} \text{ for } x = \alpha^{-i}$$

25 The symbol delay block 16 introduces a delay to the symbol data of the received codeword, to compensate for the delay through the other blocks in the decoder.

The error correction block 72 performs errata  
30 correction by XORing the delayed symbol data of the

received codeword with the correction output of the Forney block 62.

The monitor block 82 is optionally provided as a final  
5 pipeline stage within the decoder. As an additional  
check, the decoder re-calculates the syndromes over each  
sequence of symbols output by the decoder, in the monitor  
block 82. This allows mistakes in the decoder to be  
identified, and confirms that uncorrectable codewords are  
10 correctly identified.

Figure 2 shows an example circuit arranged to detect  
uncorrectable codewords. The circuit 300 is suitably  
implemented as part of the Forney block 62.

15

The detector circuit 300 comprises a logic unit 350  
that performs first through seventh logic functions (a) to  
(g), as will be described below, and an indicator unit 360  
that provides an output identifying uncorrectable  
20 codewords.

In one embodiment, the logic unit 350 comprises first  
through tenth comparators 301-310, first through fifth  
counters 311-315, and an OR block 320. The indicator unit  
25 360 comprises an output latch 361. However, it will be  
appreciated that the circuit of Figure 2 is merely one  
example and a number of alternative implementations will  
perform equivalent logic functions.

30 The detector circuit 300 identifies a codeword as  
uncorrectable if any one or more of the following  
conditions (a) to (f) are met. Condition (g) is preferred,  
but optional. Conversely, a codeword is deemed to be

correctable if none of the conditions (a) to (f) are met, or optionally none of (a) to (g).

Condition (a): No solution to key equation

5

In a first logic function it is determined whether a solution could be found to the key equation  $\sigma(x)T(x) \equiv \omega(x) \bmod x^{2T}$ . If not, then a received codeword is identified as being uncorrectable.

10

In the preferred RS decoder of Figure 1, the key equation unit 32 indicates a fail when a solution to the key equation  $\sigma(x)T(x) \equiv \omega(x) \bmod x^{2T}$  could not be found. Conveniently, in the case of key equation unit 32 based on Euclid's algorithm as described above, this failure condition is indicated by Euclid's algorithm terminating in the middle of a division step. Suitably, the circuit 300 receives a "fail" signal from the key equation block 32, as a first logic function input (a) to the OR block 320.

20

This first logic function is suitably performed as a preliminary step. Only if condition (a) is passed does the circuit need to evaluate the other conditions (b) to (g).

25

Conditions (b) to (g):

Suitably, a set of error locations and a set of erasure locations are formed. These two sets then allow conditions (b) to (g) to be evaluated quickly and conveniently. If available from elsewhere in the decoder, these sets may simply be passed to the detector circuit 300.

30

The set of error locations  $\{\alpha^{-i}, \dots\}$  is formed from the error locator polynomial  $\sigma(x)$ , by adding  $\alpha^{-i}$  to the set for each symbol  $i$  in a codeword of length  $B$  symbols where the error locator polynomial  $\sigma(\alpha^{-i})=0$  for  $0 \leq i < B$ . A variable  $n_{errors}$  is defined to be the size of this set, to represent the number of errors. The set of error locations is suitably formed by a Chien search, or equivalent. In the preferred decoder of Figure 1, the Chien search is performed by the polynomial evaluator 54. The detector circuit 300 of Figure 2 counts the variable  $n_{errors}$ , using a null detecting comparator 302 and a counter 314.

The set of erasure locations  $\{\alpha^{-i}, \dots\}$  may be formed directly or indirectly. When formed directly, the set is obtained by recalling a set of declared erasure locations, i.e. which were input to the decoder with a received codeword. However, in the example of Figure 2, the set is calculated indirectly from the erasure locator polynomial  $\Lambda(x)$ , by adding  $\alpha^{-i}$  to the set for each symbol  $i$  in a codeword of length  $B$  symbols where the erasure locator polynomial  $\Lambda(\alpha^{-i})=0$  for  $0 \leq i < B$ . A variable  $n_{erasures}$  is defined as the size of the set of erasure locations. In Figure 2, an output of the polynomial evaluator 52 is fed to a null detecting comparator 301, from which a counter 312 counts  $n_{erasures}$ .

Further, the degree (i.e. power of the highest power term) of each of the error locator polynomial  $\sigma(x)$ , the erasure locator polynomial  $\Lambda(x)$  and the errata evaluator polynomial  $\omega(x)$  is made available to the detector circuit,

such as from the polynomial evaluators 52, 54 and 56, or the Forney block 62.

Condition (b):  $\deg \sigma(x) \neq \text{nerrors}$

5

In this second logic function, the variable nerrors is compared with the degree of  $\sigma(x)$ , using the seventh comparator 307, which forms a second input (b) to the OR gate 320. If the compared values are not equal, then an  
10 uncorrectable codeword is declared.

Condition (b) addresses the case where the error locator polynomial  $\sigma(x)$  has duplicate roots. Condition (b) also addresses the case where roots of the error  
15 locator polynomial  $\sigma(x)$  indicate errors lying outside of valid locations of the codeword. Here, a shortened codeword of length  $B < 2^w - 1$  should only have errors in the shortened portion of the codeword. These two cases arise from an incorrectly formed error locator polynomial  $\sigma(x)$ ,  
20 indicating that a received codeword is uncorrectable.

Condition (c): error and erasure locations coincide

If any symbol is a member of the set of error  
25 locations and also a member of the set of erasure locations, then an uncorrectable codeword is declared. In the preferred circuit of Figure 2, an AND gate 321 drives a third counter 313 when a zero term is detected by both the null comparators 301 and 302. The output of this  
30 counter 313 is provided to a sixth comparator 306, to provide a third logic input (c) to the OR gate 320.

Condition (c) applies when the error locator polynomial  $\sigma(x)$  has a root that indicates an error lying at a location already marked as an erasure. Hence, condition (c) indicates an incorrectly formed error  
 5 locator polynomial  $\sigma(x)$ , resulting from a codeword that is uncorrectable.

Condition (d):  $\deg \omega(x) \geq \text{nerrors} + \text{nerasures}$

10 Condition (d) follows from the basic definitions of the errata locator and evaluator polynomials, in this embodiment the error locator polynomial  $\sigma(x)$ , the erasure locator polynomial  $\Lambda(x)$  and the errata evaluator polynomial  $\omega(x)$ . In particular, these polynomials show  
 15 that if there are  $\tau$  errata (i.e. errors plus erasures) in a codeword, and the codeword is correctable, then the errata evaluator polynomial  $\omega(x)$  should have a degree of at most  $\tau-1$ . Thus, if the degree of  $\omega(x)$  exceeds  $\tau-1$ , the codeword is identified as being uncorrectable.

20

In the preferred circuit of Figure 2, an uncorrectable codeword is determined when the degree of the errata evaluator polynomial  $\deg \omega(x)$  is greater than or equal to the total number of error and erasure locations. The  
 25 variable  $\text{nerasures}$  as output by the second counter 312 is added to the  $\text{nerrors}$  as output by the fourth counter 314 and the result compared to  $\deg \omega(x)$  in comparator logic 310, as a fourth logic input (d) to the OR gate 320.

30 Other possibilities are available to achieve an equivalent logic function. For example, if condition (b) is not satisfied then it can be assumed that  $\text{nerrors}$  is

equal to  $\deg\sigma(x)$ , allowing a substitution of  $\deg\sigma(x)$  for errors, if desired in a particular circuit design.

Condition (e):  $n_{\text{erasures}} + 2 \cdot n_{\text{errors}} > 2T$

5

An uncorrectable codeword is declared if the number of errors and erasures combined are greater than the guaranteed strength of the code. In this fifth logic function, an uncorrectable codeword is identified if the  
 10 number of erasures plus twice the number of errors is greater than the codeword strength  $2T$ , i.e.  $n_{\text{erasures}} + 2 \cdot n_{\text{errors}} > 2T$ . In the circuit of Figure 2, comparator logic 309 provides a fifth logic input (e) to the OR gate 320. Again, it is possible to substitute the variable  
 15  $n_{\text{erasures}}$  and  $n_{\text{errors}}$  for equivalent values  $\deg\Lambda(x)$  and  $\deg\sigma(x)$ .

Condition (f): Any symbol error location has a zero correction magnitude.

20

In the sixth logic function, an uncorrectable codeword is identified if any symbol error location has a zero correction magnitude as represented by the errata evaluator polynomial  $\omega(x)$ . That is, for every symbol error  
 25 that is located by the error locator polynomial  $\sigma(x)$ , then a non-zero correction magnitude should be provided by the errata evaluator polynomial  $\omega(x)$ . By contrast, it is possible that a symbol declared as an erasure was in fact correct, in which case the correction magnitude is zero  
 30 for the location of that symbol erasure.

In the circuit 300 of Figure 2, a fifth counter 315 is driven by an AND gate 322 in response to null determining



comparators 302 and 303 coupled to outputs of polynomial evaluators 54 and 56 respectively. Comparator logic 308 determines when the counter 315 outputs a non-zero count, to provide a sixth input (f) to the OR gate 320.

5

Condition (g):  $\deg \Lambda(x) \neq \text{nerasures}$

In this seventh logic function, an uncorrectable codeword is declared where the number of erasures is not equal to the degree of the erasure locator polynomial  $\deg \Lambda(x)$ . Typically, the erasure locator polynomial  $\Lambda(x)$  is formed directly from the declared erasures received by the decoder. It is unlikely that a correctly designed decoder will fail to generate the erasure locator polynomial  $\Lambda(x)$  correctly. Hence, this seventh logic function is optional. The fifth comparator 305 then compares nerasures with the degree of  $\Lambda(x)$ , e.g. as provided by the first evaluator 52. This comparison forms a seventh input (g) to the OR gate 320.

20

The output of the detector circuit 300 is valid once the entire codeword has been processed, as determined by a counter 311 and a comparator 304. A load signal is received to indicate that a new codeword is available at the detector circuit, and a done signal is asserted to show that the output is valid. The status output provided from the circuit 300 indicates whether an uncorrectable codeword has been determined. Ideally, the detector circuit distinguishes between correctable and uncorrectable codewords. In the example embodiment the output latch 361 indicates a correctable or uncorrectable codeword (i.e. 0=correctable, 1=uncorrectable). In other

30

embodiments, the nature of an uncorrectable error is output, e.g. responsive to the conditions (a) to (g).

5       The preferred embodiment has been described with logic operations that detect an uncorrectable codeword. In an alternate embodiment of the invention each logical test is reversed such that the detector identifies a correctable codeword, i.e. which passes the conditions (a) to (g).

10       The method and apparatus have been described which allow uncorrectable codewords to be identified accurately and efficiently in Reed-Solomon decoder handling errors and erasures. Advantageously, circuitry downstream of a Reed-Solomon decoder is informed that a severely corrupted  
15       codeword has been encountered. Appropriate action can then be taken such as requesting re-transmission or applying other data recovery techniques. When applied to a data storage medium, the detection of severely corrupted data may assist future management of the storage medium,  
20       such as avoiding physical areas of the medium responsible for the corruption. Other advantages of the present invention will be clear from the foregoing description.

## Claims

1. A method of identifying uncorrectable codewords in a Reed-Solomon decoder handling errors and erasures,  
5 comprising the steps of:

indicating an uncorrectable codeword when any one or more of the following conditions (a) to (f) is satisfied:

- 10 (a) no solution to key equation  $\sigma(x)T(x) \equiv \omega(x) \bmod x^{2T}$ ;
- (b)  $\deg \sigma(x) \neq \text{nerrors}$ ;
- (c) error and erasure locations coincide;
- (d)  $\deg \omega(x) \geq \text{nerrors} + \text{nerasures}$ ;
- (e)  $\text{nerasures} + 2 * \text{nerrors} > 2T$ ; and
- 15 (f) an error location has a zero correction magnitude;

where nerrors and nerasures represent, respectively, a number of errors with reference to an error locator polynomial  $\sigma(x)$  and a number of erasures with reference to an erasure locator polynomial  $\Lambda(x)$ ,  $2T$  is the strength of  
20 a Reed-Solomon code,  $\omega(x)$  is an errata evaluator polynomial, and  $T(x)$  is a modified syndrome polynomial.

2. The method of claim 1, comprising evaluating the  
25 condition (a) as a preliminary step, and then evaluating the conditions (b) to (f).

3. The method of claim 1, wherein the method comprises identifying a codeword as correctable if none of  
30 at least the conditions (a) to (f) are satisfied.

4. The method of claim 1, wherein the method comprises indicating an uncorrectable codeword in response to condition (g)  $\deg\Lambda(x) \neq \text{nerasures}$ .

5 5. The method of claim 1, wherein the method comprises receiving the error locator polynomial  $\sigma(x)$ , the erasure locator polynomial  $\Lambda(x)$  and the errata evaluator polynomial  $\omega(x)$ ; forming a set of error locations, and a set of erasure locations, and forming variables  $\text{nerrors}$  and  $\text{nerasures}$  representing the size of each set, respectively; and finding  $\deg\sigma(x)$ ,  $\deg\Lambda(x)$ , and  $\deg\omega(x)$ , as a degree of the error locator polynomial  $\sigma(x)$ , the erasure locator polynomial  $\Lambda(x)$  and the errata evaluator polynomial  $\omega(x)$ , respectively.

15

6. A detector circuit arranged to identify an uncorrectable codeword, for use in a Reed-Solomon decoder handling errors and erasures, the circuit comprising:

- 20 a logic unit arranged to identify each condition:
- (a) no solution to key equation  $\sigma(x)T(x) \equiv \omega(x) \bmod x^{2T}$ ;
  - (b)  $\deg\sigma(x) \neq \text{nerrors}$ ;
  - (c) error and erasure locations coincide;
  - (d)  $\deg\omega(x) \geq \text{nerrors} + \text{nerasures}$ ;
  - 25 (e)  $\text{nerasures} + 2*\text{nerrors} > 2T$ ; and
  - (f) an error location has a zero correction magnitude;

where  $\text{nerrors}$  and  $\text{nerasures}$  represents, respectively, a number of errors and erasures with reference to an error locator polynomial  $\sigma(x)$  and an erasure locator polynomial  $\Lambda(x)$ ,  $2T$  is the strength of a Reed-Solomon code,  $\omega(x)$  is an errata evaluator polynomial, and  $T(x)$  is a modified syndrome polynomial; and

30

## BEST AVAILABLE COPY

an indicator unit arranged to indicate an uncorrectable codeword, responsive to the logic unit.

5 7. The circuit of claim 6, wherein the circuit comprises a counter arranged to count nerrors and nerasures as the size of a set of error locations derived from the error locator polynomial  $\sigma(x)$ , and a set of erasure locations derived from the erasure locator  
10 polynomial  $\Lambda(x)$ , respectively.

8. The circuit of claim 6, wherein the logic unit is arranged to identify an uncorrectable codeword in response to condition (g)  $\deg\Lambda(x) \neq \text{nerasures}$ .

15 9. A method of identifying uncorrectable codewords in a Reed-Solomon decoder handling errors and erasures, substantially as hereinbefore described with reference to the accompanying drawings.

20 10. A detector circuit arranged to identify an uncorrectable codeword, for use in a Reed-Solomon decoder handling errors and erasures, substantially as hereinbefore described with reference to the accompanying  
25 drawings.

## ABSTRACT

IDENTIFYING UNCORRECTABLE CODEWORDS IN A  
 REED-SOLOMON DECODER FOR ERRORS AND ERASURES

5

In a Reed-Solomon decoder handling both errors and erasures, an uncorrectable codeword is identified when any one or more of six conditions (a) to (f) is satisfied:

10 (a) no solution to key equation  $\sigma(x)T(x) \equiv \omega(x) \bmod x^{2T}$ ;

(b)  $\deg \sigma(x) \neq \text{errors}$ ;

(c) error and erasure locations coincide;

(d)  $\deg \omega(x) \geq \text{errors} + \text{erasures}$ ;

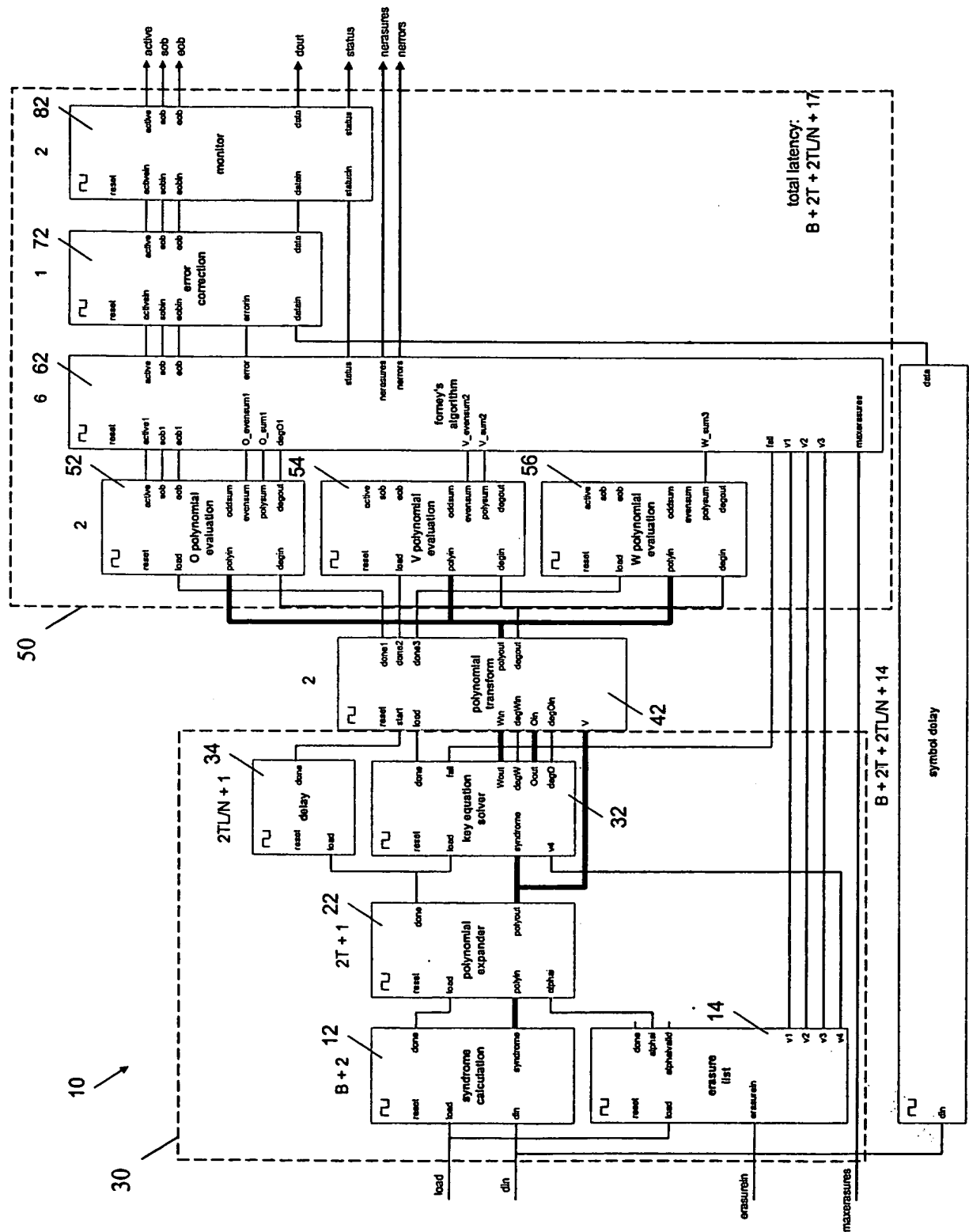
(e)  $\text{erasures} + 2 \cdot \text{errors} > 2T$ ; and

15 (f) an error location has a zero correction magnitude.

Nerrors and nerasures represent, respectively, a number of errors and erasures, with respect to an error locator polynomial  $\sigma(x)$  and an erasure locator polynomial  $\Lambda(x)$ ,  $2T$  is the strength of a Reed-Solomon code,  $\omega(x)$  is an errata evaluator polynomial, and  $T(x)$  is a modified syndrome polynomial. A detector circuit 300 comprises a  
 20 logic unit 350 which tests for the conditions (a) to (g), and an indicator unit 360 which provides a corresponding output.

25

[Figure 2]



**Fig. 1**

**THIS PAGE BLANK (USPTO**





**THIS PAGE BLANK (USPTO)**